


**PATENT
5181-35900
P4224**

"EXPRESS MAIL" MAILING LABEL
NUMBER EL690353521US
DATE OF DEPOSIT 10/5/00
I HEREBY CERTIFY THAT THIS PAPER OR
FEE IS BEING DEPOSITED WITH THE
UNITED STATES POSTAL SERVICE
"EXPRESS MAIL POST OFFICE TO
ADDRESSEE" SERVICE UNDER 37 C.F.R. §
1.10 ON THE DATE INDICATED ABOVE
AND IS ADDRESSED TO THE ASSISTANT
COMMISSIONER FOR PATENTS, BOX
PATENT APPLICATION, WASHINGTON,
D.C. 20231


Derrick Brown

**A SYSTEM AND METHOD FOR HOT SWAPPING DAUGHTERCARDS
IN HIGH AVAILABILITY COMPUTER SYSTEMS**

By:

William L. Robertson

Atty. Dkt. No.: 5181-35900

B. Noel Kivlin/EAH
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, TX 78767-0398
Ph: (512) 476-1400

005001 44208960

BACKGROUND OF THE INVENTION

1. Field of the Invention

5 This invention relates to computer hardware, and more particularly, to daughtercards which may be added to a computer system with no interruption to system operation.

2. Description of the Relevant Art

10

Computer systems that must operate for long periods of time without interruption are sometimes referred to as high availability computer systems. Many high availability computer systems are also configured to continue operation despite some system component failures. Examples of high availability computer systems may include 15 industrial computers or network file servers. In addition to the core elements of the computer system, such as the processor and memory, many high availability computer systems also include additional hardware. Such hardware may include network interface cards, process controllers, or cards providing other types of functionality. At times, it may be necessary to add or remove hardware (i.e. daughtercards) in a high availability 20 computer system. In many cases, the computer system must continue uninterrupted operation despite this need. To this end, various standards have been developed that allow hardware to be added or removed from a computer system with no interruption to system operation (often referred to as "hot swapping"). One such standard is the CompactPCI® Hotswap Specification. This standard provides guidelines for designing 25 daughtercards which may be inserted into an operating computer system without removing power or interrupting operations, as well as providing procedures for initiating software communications with the daughtercard. The standard also provides guidelines for terminating software communications between the daughtercard and computer system, as well as removing the daughtercard without interrupting operations.

00500T" 44200960

09680244-100500

A first requirement for any computer system to support hot swapping is to provide circuitry that enables the safe, orderly power-up of a daughtercard upon insertion, and orderly power-down upon preparation for removal. Daughtercards configured for hot swapping typically include power circuitry which isolates the functional circuitry (i.e. that circuitry which enables the daughtercard to perform its intended function) from system power. After detection of system power, the power circuitry may then allow power to be provided to the entire daughtercard. Some power circuitry may also be configured to control the sequence at which certain devices on the daughtercard receive power. Similarly, prior to removing a daughtercard from an operating computer system, power circuitry may remove power from the daughtercard in a controlled fashion.

A second requirement to support hot swapping is the establishment of software communications between the host computer system and the daughtercard. Software communications may be established manually through the intervention of a user of the computer system, or automatically. In the CompactPCI® Hotswap Specification, software communications may automatically be initiated following the power-up sequence by the assertion of a signal referred to as ENUM#. The ENUM# signal is bussed within a host computer system, as several daughtercards may share a common signal line. When a host computer system detects that a daughtercard has asserted the ENUM# signal, it may then begin polling the various daughtercards in order to determine which one asserted the signal. Once the host computer system determines the source of the asserted ENUM# signal, it may then begin to initiate software contact with the appropriate daughtercard.

The ENUM# signal may also be used to terminate software communications between a daughtercard and the host computer system. Daughtercards that conform to the CompactPCI® Hotswap Specification include an ejector handle for removing the daughtercard from the computer system. This ejector handle may activate a switch on the daughtercard, which may in turn assert the ENUM# signal. After determining the source

of the asserted ENUM# signal, the host computer system may then begin the process of terminating software communications with the appropriate daughtercard. Once software communications have been terminated, the daughtercard may then begin its power-off sequence. When the power-off sequence is complete, the daughtercard may then be
5 physically extracted from the computer system. Daughtercards which conform to the CompactPCI® Hotswap Specification include light-emitting diode (LED) which, when illuminated, indicate that it is safe to extract the board from the computer system.

One problem that arises with systems such as that described above pertains to the
10 use of bussed signals. As previously stated, the ENUM# signal of the CompactPCI® Hotswap Specification is a bussed signal. This is illustrated in Figure 1. Within the host computer system, the signal path for the ENUM# signal is common to each daughtercard slot (i.e. connector). Thus, when the ENUM# signal is asserted, the host computer system must first determine from which slot a daughtercard has asserted the signal. This
15 may require polling each of the daughtercards in order to make this determination. Such polling may add complexity to system software routines for controlling the hotswap process. Furthermore, the computer may be required to use several bus cycles in order to determine which daughtercard asserted the signal. The use of the extra bus cycles may have an adverse affect on system performance.

20

SUMMARY OF THE INVENTION

The problems outlined above may in large part be solved by a system and method
25 for hot swapping daughtercards in high availability computer systems in accordance with the present invention. In one embodiment, a high availability computer system includes a peripheral bus. Daughtercards may be added to the computer system by inserting them into connectors associated with the peripheral bus. The daughtercards are configured to allow their insertion or removal from the computer system without interruption to system

09630244-100500

operations. When inserted into a computer system, a daughtercard may be powered up by power control circuitry on the daughtercard. When the daughtercard is powered up, it may then assert a configuration change signal (such as the ENUM# signal described above). The computer system may then respond to the assertion of the configuration change signal by establishing software communications with the daughtercard. The configuration change signal may be driven to a storage unit located within a bus interface unit of the computer system. The state of the configuration change signal may be stored within a storage location of the storage unit. The storage location in which the state of a configuration change signal is stored for a given daughtercard is exclusive to that daughtercard. Thus, when a computer system detects the assertion of a configuration change signal, it may immediately make a determination as to which daughtercard asserted the signal.

In one embodiment, removal of the daughtercard from the computer system may begin with the assertion of the configuration change signal. The configuration may be asserted in response to the actuation of a switch mounted to the daughtercard. The switch may be actuated by an ejector handle attached to the daughtercard. When the computer system detects the assertion of the configuration change signal, it may then begin the termination of software communications with the daughtercard. In some cases, software communications may terminate immediately, while in others, a given task involving the daughtercard may be allowed to complete before terminating communications. When software communications have been terminated, power control circuitry may begin to remove power from the daughtercard. Following the removal of power from the daughtercard, a light-emitting diode (LED) on the daughtercard may then be illuminated. When illuminated, the LED indicates to a user of the computer system that it is safe to extract the daughtercard.

Thus, in various embodiments, the system and method for hot swapping daughtercards in high availability computer systems may allow the addition or removal of

00500T" 44208960

10

15

20

25

peripheral hardware with no interruption to system operation. The use of a separate storage location for storing the state of a configuration change signal for each daughtercard may allow for the simplification of system software. Instead of polling, the use of separate storage locations may allow the computer system to immediately
5 determine which daughtercard asserted the configuration change signal. Thus, associated polling routines may be eliminated, thereby simplifying system software.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading
5 the following detailed description and upon reference to the accompanying drawings in
which:

Figure 1 (prior art) is a block diagram of a computer system with a bussed
ENUM# signal;

10

Figure 2 is a block diagram of one embodiment of a computer system including
separate configuration change signals for each daughtercard;

Figure 3 is a block diagram of one embodiment of a daughtercard configured for
15 hot swapping;

Figure 4A is a schematic diagram illustrating the generation of the sense and
presence signals used in one embodiment;

20 Figure 4B is a schematic diagram illustrating the generation of the configuration
change signal of one embodiment;

Figure 5 is a flowchart illustrating a method of insertion of a daughtercard into an
operating computer system for one embodiment; and

25

Figure 6 is a flowchart of a method of extraction of a daughtercard from an
operating computer system for one embodiment.

00660244-100500

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and description thereto are not intended to limit the invention to the particular form disclosed, but, on the contrary, the invention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

10

DETAILED DESCRIPTION OF THE INVENTION

Turning now to Figure 2, a block diagram of one embodiment of a computer system including separate configuration change signals for each daughtercard is shown.

15 Computer system 1000 includes a central processing unit 1001 and a bus interface unit 1002 which are coupled by CPU bus 1003. Bus interface unit 1002 is also coupled to a peripheral bus 1005. In the embodiment shown, peripheral bus 1005 is a peripheral component interconnect (PCI) bus which conforms to the CompactPCI® Hotswap Specification. Other embodiments using other types of peripheral buses, such as an
20 Industry Standard Architecture (ISA) bus are possible and contemplated. Connectors 1006 are associated with interface bus 1005. Each connector 1006 allows a daughtercard 100 to be coupled to interface bus 1005. Daughtercards 100 may be one of many different types of peripheral hardware. Examples of common daughtercards include network interface cards, process controller cards, and video graphics cards.

25

Each daughtercard 100 is configured to assert a configuration change signal. In the embodiment shown, configuration change signal is asserted as a logic low voltage. When asserted, the configuration change signal may be driven to a storage unit 1004 via signal paths 1007. A separate signal path 1007 for conveying a configuration change

00500T " 44202960

signal is provided for each daughtercard 100. Storage unit 1004, which is located within bus interface unit 1002, may be configured to store a state of the configuration change signal. Storage unit 1004 includes separate storage locations each corresponding to a daughtercard 100. CPU 1001 is configured to execute instructions that allow the state
5 stored in each storage location to be periodically read from storage unit 1004. A logic low state detected in a given storage location may immediately allow CPU 1001 to determine which daughtercard 100 asserted a configuration change signal. If the daughtercard 100 asserting the configuration change signal has been freshly inserted into computer system 1000, CPU 1001 may execute additional instructions which establish
10 software communications with the daughtercard. After CPU 1001 has successfully established software communications with daughtercard 100, computer system 1001 may then utilize the additional resources which it provides, and the configuration change signal may then be de-asserted.

15 A daughtercard 100 already present within computer system 1000, with software communications established, may also assert a configuration change signal, indicating that the daughtercard is to be extracted. When this occurs, CPU 1001 may execute instructions which cause a termination of software communications between daughtercard 100 and computer system 1000. Following the termination of software
20 communications, daughtercard 100 may begin a power-down sequence. Daughtercard 100 may be safely extracted from computer system 1000 following completion of the power-down sequence.

It is noted that while bus interface unit 1002 is directly coupled to CPU 1001 via
25 CPU bus 1003 in the embodiment described above, bus interface unit 1002 may be operatively coupled to CPU 1001 through multiple buses and/or interfaces in other embodiments.

00500T " 4420960

Figure 3 is a block diagram of one embodiment of a daughtercard configured for hot swapping. Daughtercard 100 includes an interface chip 102, a power controller 103, and daughtercard functional logic 104 mounted upon printed circuit board (PCB) 101. Connector 105 is mounted to PCB 101, and is adapted to allow insertion of daughtercard 100 into computer system 1000. Daughtercard 100 also includes an ejector handle 106 and a switch 107.

Interface chip 102 provides a bridge between daughtercard functional logic 104 and a peripheral bus within computer system 1000. Interface chip 102 may transmit and receive bus signals associated with the peripheral bus from computer system 1000. Interface chip 102 may also receive a clock signal from computer system 1000. Daughtercard functional logic 104 may send signals to and receive signals from interface chip 102. In general, daughtercard functional logic 104 is circuitry which performs the primary function of the daughtercard. Daughtercard 100 may be one of many different types of daughtercards.

Power controller 103 provides power circuitry which allows for the safe and orderly power-up of daughtercard 100 following insertion into the computer system. When daughtercard 100 is inserted into computer system 1000, power is received by power controller 103 through a power pin (denoted Vcc) of the computer system. A sense signal and a reset signal may also be received by the daughtercard from the computer system. When power controller 103 receives a sense signal from computer system 1000, it may begin powering up daughtercard 100. In the embodiment shown, power controller 103 may assert a Power On signal, which may then turn on power transistor 108. When activated, power transistor 108 allows system power to be provided to interface chip 102 and daughtercard functional logic 104. Other embodiments of various daughtercards may be powered up in a different manner.

005001-100500

Power controller 103 is also configured to receive a reset signal from computer system 1000. The reset signal may be asserted during the power-up sequence for daughtercard 100. Power controller 103 may respond to receiving the reset signal by driving a bus reset signal to interface chip 102. This may result in bus signals from
5 interface chip 102 being in a reset state once interface chip 102 is receiving power, which may prevent daughtercard 100 from interacting with the peripheral bus.

When all circuitry of daughtercard 101 has successfully powered up, daughtercard 101 may then assert a board OK signal. In some embodiments, the board OK signal may
10 be asserted once the daughtercard is powered up. In other embodiments, additional conditions (such as passing a built-in self-test) may be required for asserting the board OK signal. Computer system 1000 may respond to the assertion of the board OK signal by de-asserting the reset signal.

15 Following the de-assertion of the reset signal, power controller 103 may then respond by asserting the Configuration Change signal. When asserted by a freshly inserted daughtercard, the configuration change signal indicates to computer system 1000 that a new daughtercard is present and ready for initiation of software communications. In response to detecting the asserted Configuration Change signal, computer system 1000
20 may then begin establishing software communications with daughtercard 100. When computer system 1000 has successfully established software communications with daughtercard 100, the Configuration Change signal may be de-asserted.

Daughtercard 100 also includes an ejector handle 106 (some embodiments may
25 have multiple ejector handles). In the embodiment shown, ejector handle 106 may be used to actuate switch 107. When actuated, switch 107 provides an indication to daughtercard 100 that it is about to be extracted from the connector of computer system 1000 into which it is coupled. Power controller 103 may respond to the actuation of switch 107 by asserting the Configuration Change signal. In this case, computer system

1000 may respond to the assertion of the Configuration Change signal by terminating software communications with daughtercard 100. In many cases, computer system 1000 will allow daughtercard 100 to complete any pending operations before terminating software communications. Once software communications have been terminated, 5 computer system 1000 may assert the reset signal in order to prevent daughtercard 100 from interacting with the peripheral bus. Computer system 1000 may also de-assert the sense signal when software communications are terminated. Power controller 103 may then begin a power-down sequence, removing power from interface chip 102 and daughtercard functional logic 104. With software communications terminated and the 10 power-down sequence completed, power controller 103 may then illuminate light-emitting diode (LED) 108. When illuminated, LED 108 indicates to a user that daughtercard 100 may be safely extracted from the computer system.

Moving now to Figure 4A, a schematic diagram illustrating the generation of the 15 sense and presence signals used in one embodiment is shown. The sense signal line in this embodiment is terminated by resistor 112B, which serves as a weak pull-down resistor, keeping the signal line at a logic-low voltage (near ground potential). When the daughtercard is inserted into the computer system, a strong pull-up resistor 112A overrides the weak pull-down resistor, and thus the sense signal is asserted as a logic-high 20 voltage. The sense signal is then received by power controller 103 on the daughtercard, and by the input to amplifier 110A of the computer system. Amplifier 110A is configured to respond to the logic-high input by driving the presence detect signal to a logic-high on its output. The presence detect signal may then be used by the computer system to detect the presence of a freshly inserted daughtercard. In order to de-assert the 25 sense signal when a daughtercard is connected, power is provided to amplifier 110B through the signal line labeled Power On (amplifier 110B is normally off, with no power applied). Since the input of amplifier 110B is connected to ground, the output of the amplifier (when operating) will be at or near ground potential. The sense signal may be

deasserted in this manner when it is necessary to remove the daughtercard from the computer system.

In Figure 4B, a schematic diagram illustrating the generation of the configuration change signal of one embodiment is shown. When no daughtercard is connected, the input signal trace to storage unit 1004 is held at a logic-high voltage by pull-up resistor 114. When the daughtercard is connected, the logic-high state will remain as long as transistor 115 is turned off. Transistor 115, arranged in an open collector configuration, may be turned on by power controller 103. The configuration change signal may be asserted when power controller 103 turns on transistor 115. When transistor 115 is on, it may pull down the voltage present on the signal line to a logic-low voltage, overriding the pull-up resistor. Thus, the configuration change signal is asserted in a logic-low state in this embodiment. Storage unit 1004 may then store the logic-low state, thereby allowing the computer system to detect the assertion of the configuration change signal. A reading of a logic-low state from the storage unit may indicate to the computer system that a daughtercard has been recently inserted into a connector of the peripheral bus and is ready for establishment of software communications. A reading of a logic-low state may also indicate that a daughtercard is to be extracted from the computer system in cases where the daughtercard is present with software communications already established.

Turning now to Figure 5 a flowchart illustrating a method of insertion of a daughtercard into an operating computer system for one embodiment is shown. The method begins with step 2001, as a daughtercard is inserted into a computer system. Upon insertion of the daughtercard, sense and presence detect signals are asserted. The presence detect signal provides an indication to the computer system that a daughtercard has been freshly inserted into the system. The sense signal provides an indication to the daughtercard that its presence has been acknowledged by the computer system. After receiving the sense signal, a power controller on the daughtercard may then begin a power-up sequence. If, by step 2002, the daughtercard has successfully powered up, a

board ok signal will be asserted by the daughtercard and the computer system will de-assert the reset signal, as in step 2004.

After verifying that the reset signal is de-asserted and the board ok signal is asserted (in step 2006), the daughtercard may then assert the configuration change signal in step 2008. The assertion of the configuration change signal is verified in step 2010. After the assertion of the configuration change signal, the computer system may respond by initializing software communications with the daughtercard. Establishing software communications may include such tasks as loading drivers into memory and allocating various system resources. After verifying that software communications have been fully established (step 2014), normal daughtercard operations may be commenced. At this point, the computer system may employ those resources provided by the daughtercard.

If, in step 2002, the daughtercard fails to fully complete its power-up sequence, the power controller on the daughtercard may respond by performing a power-down sequence in step 2003. Similarly, if the board ok signal is not asserted by the daughtercard, or the reset signal is not de-asserted by the computer system (step 2006), the power controller may begin performing the power-down sequence of step 2003. The power-down sequence of step 2003 may also be performed if the configuration change signal is not asserted by the daughtercard (step 2010).

If the initial attempt by the computer system to communicate with the daughtercard is unsuccessful (step 2014), the computer system may make further attempts at establishing software communications (step 2005). In some embodiments, subsequent attempts by the computer system to communicate with the daughtercard may occur automatically, while in other embodiments operator intervention may be required. If repeated attempts at communicating with the daughtercard fail, the power-down sequence may then be performed (step 2003). When the power-down sequence has completed, the daughtercard may then be physically extracted from the computer system (step 2007).

